《研究論文》

Journal of the Human Engineering Society of Korea
Vol.6 No.2,Dec. 1987

컴퓨터 그래픽 User Interface 설계에서의 Human Factor

(The Human Factors in User Interface Design of Computer Graphics)

崔 潤 哲†

Abstract

This paper discusses the gereral principles to be considered in the design of user interfaces of graphics packages and presents a top-down design process in systematic way.

The effective and convenient user interfaces are analyzed based on human factors criteria and we discuss the properties and application requirements of typical interaction techniques which support primitive tasks. The choice of an interaction technique has a set of input device prerequisites to be met.

I. 서 론

대화식 그래픽스(interactive graphics)의 목적은 사용자와 컴퓨터간의 효율적인 대화(communication) 수단을 통하여 그래픽작업을 처리하는 데 있다. 따라서 대화식 그래픽 패키지 설계시 단순히 그래픽과정을 수행할뿐 아니라 사용자가 어떠한 방식으로 수행하는지를 고려해야 한다. User interface는 그래픽 패키지 사용자가 목적물을 디스플레이하고 속성을 지정하고 기하학적 변환(transformation)등의 기본적 그래픽기능을 수행할 때 가장 편리하고 효율적인 방식으로 처리할 수 있도록 지원하여야 한다. 각 응용분야에 따라 가장 적절한 대화식방식을 사용자 입장에서 설계하고 이를 효율적으로 지원할 수 있는 interactive 기법과 입력장치를 선택한다. 잘못 설계된 user interface는

† 연세대학교 전산과학과

사용자로 하여금 작업수행을 하는데 필요없이 많은 시간을 소모케하며 과오를 야기시킬 여지를 증가시킨다.

이러한 이유에서 user interface를 설계할때는 human factor를 고려하여야 한다[1,2]. 그럼에도 불구하고 과거에는 컴퓨터 연구에서는 human factor가 강조되지 않았다. 그 이유는 CPU time과 memory space의 최적이용 방법이 과거의 주요관심사이었기 때문이다. 그러나 최근 하드웨어 가격이 급격히 하락하고 있고 강력한 그래픽 기능을 제공하는 퍼스널 컴퓨터나 워크 스테이션이 확산됨에 따라 컴퓨터 하드웨어 측면의 효율성보다 사용자의 효율성이 더욱 중요하게된 시점에 이르렀다. Human factor측면에서 보면 좀 더 많은 컴퓨터 자원을 사용하더라도 이용자 생산성향상과 편리에 도움이 되는 방향으로 user interface를 설계해야 한다[1,3].

User interface의 설계가 체계적이고 과학적 요소보다 art적 요소가 많기 때문에 아직까지 이 분야의 체계적 연구가 부족하였다[2].이 분야의 연구는 computer science뿐만 아니라 perceptual psychology, cognitive psychology, human factor분야 등에서 각기 분산적이고 부분적으로 수행되어 왔다. 본 논문은 지금까지의 경험, 실험결과, human factor문헌 등을 종합하여 대화식 user-computer interface 설계에 적용할 수 있는 interactive기법과 물리적장치를 체계적인 방식으로 분석하여 설계자가 적용할 수 있도록 제시한다.

Ⅱ. User interface의 설계원칙

User interface 설계시 human factor 측면에서 바람직한 설계원칙을 제시한다. 이러한 설계원칙들을 그대로 적용한다해서 성공적인 user interface를 보장하는 것은 아니지만 훌륭한 설계를 위해서는 반드시 고려되어야 한다[1,4].

1) Feedback 방식

Feedback이란 user input에 대한 응답(response)을 의미한다. 사용자에게 메시지를 주거나 메뉴가 선택되었을 때 신호를 보냄으로써 명령어가 수신되었음을 사용자가 인식함으로써 그래픽 시스템 사용에 도움을 준다. 시스템은 그래픽처리의 각 과정에서 필요할 때마다 적합한 feedback을 출력함으로써 시스템이 현재 어떤 작업을 수행중인지 알 수 있다. 특히 응답시간이 긴 경우 feedback이 더욱 요구된다. Feedback은 분명하되 user의 집중을 혼란하게 해서는 안된다.

자주 이용되는 feedback 방식을 논하기로 한다. Function key를 누를때 소리나 불빛을 발생시킴으로 사용자에게 feedback을 제공할 수 있다. 메뉴에서 command를 선택할때 raster디스플레이 장치에서는 선택된 command를 포함하는 공간의 pixel intensity를 역으로 디스플레이함으로써 사용자가 쉽게 인식할 수 있다. 스크린상에 메시지를 디스플레이 할때는 항상 일정한 위치에 디스플레이하는 방식과 cursor 부근의 user work area에 feedback 메시지를 디스플레이 하는 방식이 있다. 경우에 따라서는 feedback을 보낼때 심볼을 이용하는것이 바람직하다. 예를들면 현재 수행중인 경우에 blinking심볼을 디스플레이하고 에러가 발생했을때 아래로 향한 엄지손가락 심볼을 디스플레이 한다. 때로는 360° dial상에 움직이는 바늘을 디스플레이 함으로써 몇 퍼센트의 작업이 현재 수행되었는지를 보여줄 수도 있다.

2) 사용자 HELP 기능

Command language가 HELP기능을 소유하는 것은 매우 중요하다. 여러 수준의 HELP를 제공함으로써 초보자에게는 각 과정에서 자세한 Instruction이나 prompt를 주고 경험이 많은 사용자에게 간단한 prompt를 보여줌으로써 작업집중에 방해가 되지 않도록 한다. 때로는 HELP기능이 tutorial session을 포함할 수도 있다. 초보자에게는 여러가지 경우에서 각종 command가 어떻게 이용되고 어떤 응용작업을 수행할 수 있는지 보여줄수도 있다.

3) 기억의 최소화

Command language가 사용자에게 이해되기 쉽고 command 이용목적이 기억하기에 용이해야 한다. 불분명하거나, 복잡하거나, 일관성이 없거나, 너무 단축된 command 형식은 사용자에게 혼돈을 주거나 그래픽 패키지의 효율성을 저하시키므로 피해야 한다. 사용자가 command를 이용할 때 한가지 입력장치에서 다른 입력장치로 빈번하게 옮기며 사용하지 않도록 command가 조직되어야 한다. 또한 다양한 사용자 계층을 지원하기 위하여 command language 가 여러 level로 설계되어야 한다. 즉, 초보자는 작은 수의 command 집합을 이용하여 기본적인 작업을 수행하되 전문가는 고차원적인 command들을 이용하여 복잡한 작업을 수행하게 된다.

4) Backup과 Error handling

Command의 수행도중 backup 또는 중단(aborting)할 수 있는 장치가 존재해야 한다. 자주 command의 수행이 완결되기 전에 command가 취소되기도 한다. 이 경우 시스템은 command가 전혀 시작되기 이전의 상태로 환원되어야 한다. 어떤 시점에서든지 backup할 수 있다면 사용자는 실수의 결과에 신경쓰지 않고 시스템 기능을 자유자재로 이용할 수 있게 된다. 때로는 그래픽에서 어떤 목적물의 최종위치를 정하기 전에 여러가지 다른위치와 비교함으로써 시스템에 매우 유용한 backup을 지원한다. 또한 사용자가 에러를 일으켰을 때 그 이유를 즉시 알아볼 수 있도록 적절한 에러 메시지와 진단을 출력할 수 있어야 한다.

5) 응답시간(response time)의 제어

Command의 복잡도에 관계없이 사용자는 시스템이 즉시 응답하기를 기대한다. 그렇지 않으면 사용자는 시스템이 작업 수행중인지 아닌지를 알 수 없게 된다. 만일 처리작업이 복잡하여 즉시 처리 결과를 출력할 수 없을시에는 어떤 1차응답을 즉시 디스플레이 하는 것이 바람직하다. 훌륭한 그래픽 패키지는 사용자 입력을 받은후 1/10초 이내에 이러한 응답을 기대한다. 일반적으로 때로는 빠르고 때로는 매우 늦은 응답보다는 일정하게 항상늦은 응답이 사용자에게 좌절감을 주지 않는다.

6) 일관성(consistency)있는 설계

그래픽 시스템의 작동방식이나 command language가 획일적으로 조직되어 있어 예외나 특수 경우를 허용하지 않는것이 바람직하다. command language에서 서로 다른 command 구조를 최소화함으로써 보다 일관성있게 이용할 수 있고 기억하기가 쉽고 에러를 야기시킬 확률이 줄어든다. 또한 어떤 command가 한 부류의 목적물(object)을 처리할 수 있을때 유사한 다른 부류의 목적물에도 이 command가 같은 방식으로 적용될 수 있어야 한다. 컴퓨터로 부터 처리결과의 출력도 유사한 경우에는 같은 위치에 논리적으로 동일한 방식으로 디스플레이 되어야 한다. 시스템이 일관성이 있으므로해서 시스템 사용자의 한 영역에서의 시스템 이용에 관한 지식과 경험이 다른 영역에서도 적용될 수 있게 된다. 일관성있는 설계를 위하여 시스템의 top-down 설계방식이 자주 이용된다.

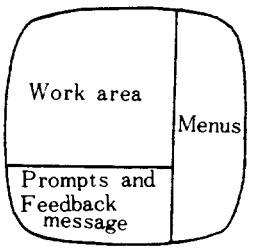
7) 메뉴(menu)의 설계

대부분의 그래픽 패키지는 메뉴를 이용한다. 메뉴를 이용함으로써 사용자는 광범위한 input option(command, object, parameter, attribute등)의 기억부담을 덜어준다. 또한, 다른 응용을 위해서 function key나 button을 이용하는 경우에는 프로그램을 다시하고 레벨을 다시 붙여야 하는데 반하여 메뉴는 쉽게 변경될 수 있는 이점이 있다. 메뉴를 작은 submenu들로 조직하는 방법의 하나는 메뉴가 계층적(hierarchical)구조를 갖도록 하는 것이다. 일반적으로 2~3계층보다 큰 계층구조는 사용자에게 혼란을 초래하므로 피해야 한다. 메뉴에 표시되는 항목은 character string이나 icon(geometric shape)이다. Icon은 상대적으로 적은

공간을 차지하고 character string보다 빨리 인식될 수 있는 장점이 있다[5]. 대부분의 응용에서 메뉴는 항상 스크린의 같은 장소에 디스플레이 된다.

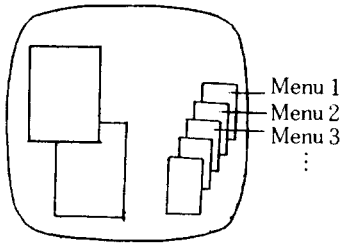
8) Output format

그래픽 패키지는 사용자에게 출력결과의 그림, 메뉴, 메시지, 기타 다른 정보를 디스플레이 한다. 시스템 설계자는 이러한 정보가 사용자에게 시각적 측면에서 가장 효율적으로 이용될 수 있도록 스크린을 할당해야 한다. 스크린 배치의 가장 기본적인 요소는 사용자 작업장소(user work area), 메뉴장소, prompt와 feed back 메시지를 디스플레이하는 장소이다. 일반적으로 이러한 장소는 스크린상에 고정된 위치를 차지하는 것이 바람직하다.



(그림 1) 대표적인 screen layout

가장 대표적인 스크린 구도가 그림 1에 나타나 있다. 스크린 구도를 보다 융통성 있게 구성하는 방법은 필요한 수 만큼의 overlapping window를 스크린 상에 디스플레이 하는 것이다. 이 방식에서 사용자는 그림을 디스플레이 할 work area와 현재 active한 메뉴를 지정해 주어야 한다(그림 2). 그래픽 패키지가 Zooming이나 panning 기능을 가짐으로써 디스플레



(그림 2) overlapping window를 가지는 screen layout

이된 그림의 일부를 확대하거나 원하는 그림의 부분을 이동하여 볼수 있게 된다.

Ⅲ. 설계과정(Design process)

대화식 그래픽 응용프로그램의 user-computer interface 설계의 첫 단계는 응용분야와 사용자집단을 이해하는데 있다. 이러한 과정을 task analysis 또는 requirement definition이라 부른다. Task analysis를 가장 성공적으로 수행하기 위한 한가지 방법은 실제 평소 사람이 응용분야의 작업을 어떠한 방식으로 처리하는지 관찰하는데 있다. Task분석이 끝난후 아래의 설계과정을 통한 top-down 설계방식을 따른다[1].

1) Conceptual design(User model)

Conceptual model에서는 사용자에게 제공할 기능의 일반적 형태를 정의한다. 이 경우 사용자가 디스플레이 할 목적물(object)의 종류와 목적물에 적용할 작업(operation 또는 action) 등이 정의된다. 즉 conceptual model은 시스템이 어떤 목적을 성취하기 위하여 설계되고 이 목적을 달성하기 위하여 어떤 object와 어떤 operation이 요구되는지를 개념적으로 규명하여 하나의 모델을 만든다. 이때 목적물은 가구, 논리회로의 게이트 등과 같은 application object와 cursor, menu selection symbol, positioning grid와 같은 control object를 포함한다. Conceptual model은 효율적이고 일관성 있게 조직되어야 한다.

2) Syntax design

Input command의 문법적구조(syntax)를 이 과정에서 설계한다. Command는 컴퓨터로 하여금 어떠한 작업을 처리하도록 지시한다. Command의 문법적 구조는 여러가지 다양한 형태로 존재할 수 있으며 syntax에 따라 효율성이 달려있다. 간단한 예로 drafting 프로그램에서

"move entity" command는 아래의 6 가지 syntax의 형태를 취할 수 있다.

⟨move command⟩: :=⟨move⟩⟨entity⟩⟨position⟩
⟨move⟩⟨position⟩⟨entity⟩
⟨entity⟩⟨move⟩⟨position⟩
⟨entity⟩⟨position⟩⟨move⟩
⟨position⟩⟨move⟩⟨entity⟩
⟨position⟩⟨entity⟩⟨move⟩

여기서 ⟨move⟩, ⟨entity⟩, ⟨position⟩ token은 primitive nonterminal symbol 들이다. 각 primitive nonterminal symbol은 lexical design과정에서 특정한 interaction기법을 적용하여 하드웨어 입력장치에 의하여 처리된다.

3) Lexical design

이 설계는 특정한 하드웨어 장치의 기능을 input 또는 output language의 token에 부여(binding) 하는 과정을 의미한다. Input language에서는 input 장치를 이용하여 token에 해당하는 action을 수행함을 의미하고 output language의 token은 line, character, curve 등의 display primitive 들을 의미한다. 따라서 syntax design 과정이 특정 입출력기기로 부터 독립적(device independent)인데 반하여 lexical design 과정은 입출력기기에 의존한다(device dependent). 5절에서는 lexical level 에서의 interaction 기법을 논하기로 한다.

User interface설계 경험에 의하면 아무리 주의깊게 설계된 user interface도 처음부터 완벽할 수는 없다. 따라서 user interface부분은 모듈화(modularized) 되게 조직되어야 하며 특히 lexical 또는 syntax level의 설계변화에 융통성있게 순응해야 한다. 이러한 human factor의 fine tuning이 훌륭한 user interface설계에 절실히 요구된다.

Ⅳ. Human factor 의 판단기준(Ergonomic measures)

바람직한 user interface의 설계는 사용자가 작업을 수행하는데 있어서 tool(graphic terminal, 입력장치등) 사용에 최소의 의식적주의(minimal conscious attention)를 기울이고 최대효과(maximal effectiveness)를 달성하는데 있다. 이러한 목적을 달성하기 위하여 여러가지 요소가 관련된다. 본 논문에서는 user interface의 lexical level에 대하여 논하기로 한다. 즉 사용자가 처리할 기본적인 작업(task)을 위하여 어떠한 interaction 기법을 선택하며 이 기법을 적용하기 위하여 어떤 입력장치를 사용하는지 설명한다. Interaction기법의 질(quality) 은 아래의 열거된 human factor적 단서에 의하여 결정된다[2].

- Task를 수행하는데 걸리는 시간(time)
- Task를 얼마나 정확하게 수행할 수 있는지 여부(accuracy)
- 쾌락의 정도(pleasurability)
- Learning time과 recall time
- 기억부담(memory load)
- 에러 발생도(error susceptibility)와 피로감(fatigue)
- 기법의 편리한 정도(convenience)

작업수행에 걸리는 시간은 perceptual process, cognitive process, motor process 에 요하는 시간을 포함한다. 쾌락의 정도는 자발적 환경하에서는 매우 중요한 요소이나 순수한 production환경하에서는 그다지 중요한 단서가 아니다. Learning time은 사용자가 작업을 배우는데 걸리는 시간으로 다시 perceptual learing time, cognitive learning time, motor learning time을 포함한다. Recall time은 사용자가 시스템을 일정한 기간동안 이용하지 않다가 다시 이용할때 시스템 사용에 능숙해지기까지 소요되는 시간을 의미한다. 기억부담은 어떤 task를 수행하기 위하여 기억해야할 사항과 과정들을 의미한다. 작업수행상의 피로감(perceptual fatigue, cognitive fatigue, motor fatigue)은 에러발생율을 증가시키고 사용자의 만족감을 경감시킨다. Interaction 기법이 사용하기에 얼마

나 자연스러운지(naturalness)여부와 입력장치의 작업공간의 크기가 기법의 편리함에 영향을 미친다. 예를들면 어떤 operation을 자연스럽게 나타내는 icon은 매우 사용하기에 편리하다.

이러한 interaction 기법들을 동시간에 적용할 다른 기법들과 완전히 독자적으로 선택해서는 안된다. 즉 사용자가 어떤기법을 사용할 때 시간상으로 이 기법의 전후에 사용되는 기법의 효율성을 판단하여야 한다(the effect of context). 이때 전후 연관된 기법들은 상호 동일한 입력장치를 사용함이 바람직하다.

V. 논리적 기본 task를 지원하는 interaction 기법

각 interaction 기법은 lexical level에서 primitive nonterminal에 해당하는 작업을 수행한다. 이때 적절한 interaction 기법의 선택은 응용요건(application requirement)에 따라 결정된다. Interaction 기법은 고유의 특성(property)을 가지고 있다. 뿐만아니라 interaction 기법은 특정 하드웨어가 선행조건(hardware requirement)이 되는 경우가 많다. Interaction 기법의 선택을 위하여 사용자 측면(user-oriented)에서 볼때 task를 Select, Position, Orient, Quantify, Text, Stretch, Sketch, Manipulate, Shape의 9가지 기본적인 interaction task로 나누어 볼수 있다[2]. 각 기본 task는 이론적으로 어떤 물리적 입력장치(physical input device)에 의해서도 시뮬레이트(simulate)될 수 있다[1] 본 논문에서 고려할 주요한 입력장치는 tablet(with stylus), mouse, joystick, trackball, keyboard(with cursor control key), light pen, touch panel, programmable function key, voice recognizer, potentiometer등이다. 어떤 interaction 기법을 적용하고 어떤 특정한 입력장치를 사용할 것인지는 아래에 나타나 있는 여러가지 단서와 human factor 차원에 근거하여 결정 하여야 한다[6].

1) Select

사용자가 대안집합(a set of alternatives)에서 하나를 선택한다. 대표적인 interaction 기법은

(1) Light pen을 이용한 메뉴의 select.
(2) Tablet이나 mouse의 cursor를 이용한 메뉴의 select.
(3) Entity 이름을 keyboard로 직접 타이프.
(4) Touch panel 상에서 지적
(5) Programmable function key의 이용.
(6) Voice input

Select 기법의 선택은 대안집합의 크기, 범위등의 응용요건을 고려해야 한다.

메뉴의 선택시 light pen이나 touch panel을 이용하면 직접 select하는것이고 mouse나 tablet을 이용하는 것은 간접 select 방식이다. 메뉴 select는 주로 command select가 많으나 operand select(object select) 또는 attribute select도 가능하다. 메뉴의 조직은 single-level이나 hierarchical 구조를 가지고 실험결과에 의하면 2~3 level 정도를 가지는 broad한 메뉴가 select 시간이나 정확도 측면에서 이상적이다. Hierarchical 메뉴 방식을 이용할 때도 경험이 많은 사용자를 위하여 keyboard나 programmable function key에 의한 select 기법을 동시에 제공하는 것이 바람직하다. 메뉴의 각 항목은 알파베트 순서를 배열하거나 유사한 항목들을 그룹으로 모아두는것이 select시간을 단축하는데 도움이 된다. Icon을 이용한 메뉴 방식이 object menu에서 특히 바람직하고 command menu에서도 icon이 작업을 잘 묘사할 수 있는 경우에는 쉽게 인식 될 수 있는 장점이 있다. 이러한 icon menu방식은 Lisa, Macintosh, Star컴퓨터에서 자주 이용된다. 최근의 실험결과에 의하면 mouse가 joystick이나 cursor control key 보다 select 기능이 우수하다. 또한 경험이 많은 사용자에게는 mouse가 light

pen보다 성능면에서 앞서나 초보자에게는 light pen이 오히려 다소 우수하다. 이러한 이유에서 일반적으로 joystick은 select 장치로 이용되지 않는다.

2) Position

사용자가 디스플레이 장치상의 특정위치를 지정한다. 대표적인 Interaction 기법은

(1) Tablet, mouse, joystick으로 제어하는 cursor의 이용

(2) 특정위치의 좌표를 직접 타이프

(3) Light pen이나 tracking cross의 이용

Positioning기법은 차원(1-D, 2-D, 3-D), open loop(시각적 feedback이 중요하지 않은 경우) 또는 closed loop(연속적인 시각적 feedback이 요구되는 경우) 여부, 해상도(resolution) 정도에 따라 선택된다.

Positioning 시에 input device 좌표계에서 screen 좌표계로의 mapping이 회전(rotation) 없이 변환되어야 한다. Positioning 위치를 나타내는 cursor의 모양이 배경이나 주위 심볼들과 판이하게 달라 식별이 용이해야 한다. C/D ratio 즉(movement of hand)/(movement of cursor)는 스크린상의 cursor의 이동거리에 대한 입력장치의 locator의 이동거리를 나타낸다. C/D ratio가 낮을수록 빨리 cursor를 이동시킬 수 있는 반면 C/D ratio가 높아야 미세한 위치선택이 용이하다. Tablet, mouse, trackball 등은 스크린상에 연속적인 feedback을 제공함으로써 사용자는 스크린상에 원하는 위치를 대화식으로 선택할 수 있다. 이에 반하여 keyboard를 이용하는 이산적인(discrete) feedback 방식은 미리 원하는 좌표를 알때 적절한 기법이다. Tablet, mouse, trackball 장치등이 간접 positioning 방식인데 비하여 light pen장치는 직접 positioning방식이다. 간접 positioning방식을 이용하면 상대적으로 피로가 경감되지만 hand-eye coordination이 문제가 된다. 그러나 경험이 축적되어감에 따라 hand-eye coordination면에서 큰 진전을 기대할 수 있음이 실험적으로 증명되었다.

3) Orient

사용자가 2차원 또는 3차원 공간상에서 디스플레이 될 entity의 방향을 결정하는 task이다. 대표적인 interaction 기법은

(1) Dial이나 joystick을 이용한 방향의 제어

(2) Keyboard를 이용하여 각도를 직접 타이프

(3) Tablet이나 mouse를 이용하는 방법

Orientation 기법은 degree of freedom, 해상도 등에 따라 결정된다.

일반적으로 목적물의 중심을 이용하여 회전시키는 것이 가장 편리한 방법이다. 3차원 공간에서는 방향을 스크린상에서 옳바르게 인식하기가 어려우므로 gnomon과 같은 시각적 수단(visual aid)이 필요하다. Positioning기법 에서와 마찬가지로 continuous feedback 기법과 discrete 기법 간에 human factor적 차이가 있다.

4) Quantify

사용자가 원하는 값의 크기를 입력하는 task로 아래의 대표적인 기법을 들 수 있다.

(1) Keyboard를 이용하여 값을 타이프

(2) Potentiometer의 사용

(3) Tablet, mouse 등의 locator 장치 이용

Quantify기법의 선택은 해상도정도, open loop 또는 closed loop 여부등에 달려있다. Potentiometer나 tablet, mouse 등을 이용할때 입력치를 나타내기 위한 scale이 디스플레이 화면에 나타나고 현재의 입력치가 scale 상에 continuous feedback을 보여주는 방식이 바람직하다.

5) Text

사용자가 text string을 입력하는 task로

(1) Keyboard를 이용한 직접입력

(2) 메뉴로부터 character string의 선택

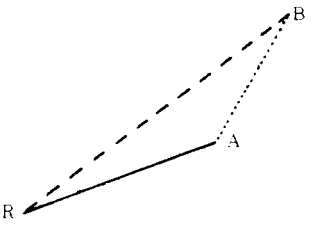
등이 대표적인 기법이다. 이외에도 voice rec-

ognizer나 character recognizer를 이용하는 방식도 있다. 그러나 많은 양의 input text를입력시킬 때는 keyboard를 이용하는 것이 automatic scanner 를 제외하고는 가장 빠른 방식이다. Voice recognizer를 이용하면 속도는 비교적 늦으나 손을 자유롭게 해주는 효과가 있다.

6) Stretch

사용자가 특정한 점(point)을 옮김으로써 이 점을 포함하는 목적물(object)의 모양을 연속적으로 변경해 볼 수 있다. 대표적인 stretching 기법은

(1) Stretched line(rubber-banding기법) (그림 3)

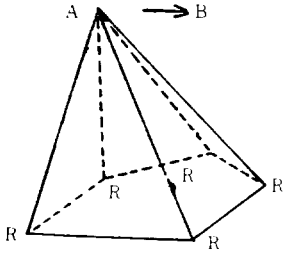


(그림 3) rubber-band line

(2) 공통점을 소유하는 line들의 stretch

(3) 수직 및 수평적인 stretch(zig-zag 기법)

(4) stretched polygon, pyramid등(그림 4)



(그림 4) rubber pyramid

Stretch기법은 positioning 기법과 동일한 응용요건과 선행조건을 가지며 필히 연속적인 시각적 feedback이 요구된다.

7) Sketch

사용자가 붓이나 펜을 이용하는것과 같은 "freehand sketching" 방식으로 곡선을 따라 이동한다. 일반적으로 tablet, mouse, joystick 등을 이용하여 연속적인 시각적 feedback을 보여주는 closed loop 방식이 적절하다. Sketching기법의 선택은 차원(dimensionality), 해상도 정도, sampling 방식(time sampling 방식 또는 distance sampling), smoothing 방식에 따라 결정된다. 곡선을 부드럽게 디스플레이하기 위해서는 일정시간 간격으로 sampling 하는 time sampling 방식이 이상적이다. Sketching 기법에서도 hand-eye coordination이 매우 중요한 과제이다. 실험결과에 의하면 tablet과 stylus를 이용하는 것이 mouse를 이용하는 것보다 성능이 우수하다.

8) Manipulate

어떤 목적을 2차원 또는 3차원 공간내에서 이동(translation), 회전(rotation), 또는 확대/축소(scaling) 시킨다. 각 경우의 manipulation 기법을 dragging, twisting, scaling 기법이라 부른다. 이 기법은 일반적으로 연속적인 시각적 feedback을 요구하며 응용요건은 positioning 기법 및 orientation 기법과 동일하다.

9) Shape

사용자가 positioning 장치를 이동시킴(즉, control point의 이동)에 따라 곡선이나 곡면 등을 매끄러운 모양으로 유지하며 변화시킨다. Control point를 이용하는 방법에 따라 Bezier 방식과 Spline 방식이 있다. 원하는 모양의 곡선이나 곡면이 생성될때 작업이 종료된다.

Ⅵ. 결 론

최근 하드웨어에 비해 소프트웨어가 더욱 중요하게 인식되고 있고 특히 모든 소프트웨어는 사용자 입장에서의 효율성과 편리함을 최우선적으로 지원할 수 있도록 설계되어야 한다. 이러한 이유에서 소프트웨어가 그래픽을 이용한

user interface를 제공한다는 것은 매우 바람직하다. 본 논문에서는 그래픽 패키지의 user-interface설계에서 고려되어야 할 일반적인 원칙을 제시하고 그 설계과정을 체계적으로 논하였다. 먼저 task analysis를 통한 conceptual model을 설계하고 이로부터 구체적인 interaction기법을 논하는 lexical model에 이르기까지 top-down방식의 설계기법이 제시되었다.

Interaction 기법의 선택은 응용요건(application requirement)과 기법의 특성(property)에 의거하여 결정될 문제이다. 특히 interaction 기법은 perceptual, cognitive, motor process 등의 human factor 측면에서 고려되어야 한다. Interaction기법은 전후에 순차적으로 이용되는 기법과의 연결성과 일관성을 고려하여 문맥적(context) 차원에서 선택되어야 한다. 따라서 user interface는 초기 시스템을 추후 실험 및 사용을 거쳐 보강해 나갈 수 있도록 모듈식 설계기법에 의해 설계되어야 한다.

參 考 文 獻

〔1〕 Foley, J. and A. VanDam, *Fundamentals of Computer Graphics*, Addison-Wesley, 1984.

〔2〕 Foley, J., V. Wallace and P. Chan, "The Human Factors of Computer Graphics Interaction Techniques," *IEEE Computer Graphics and Applications*, Vol. 4, November, 1984.

〔3〕 Good, M. D., et al, "Building a User-derived Interface", *Comm. of the ACM*, Vol. 27 (10), 1984.

〔4〕 Hearn, D. and M. Baker, *Computer Graphics*, Prentice-Hall, 1986.

〔5〕 Marcus, A., "Corporate Identity for Iconic Interface Design" The Graphic Design Perspective", *IEEE Computer Graphics and Applications*, Vol. 4, December., 1984.

〔6〕 Swezey, R. W. and E. Davis, "A Case Study of Human Factors Guideline in Computer Graphics", *IEEE Computer Graphics and applications*, Vol. 3, November., 1983.